A Fast Solver for Systems of Reaction-Diffusion Equations

M. Garbey,[1] H. G. Kaper,[2] and N. Romanyukha[3]

1 Introduction

In this paper we present a fast algorithm for the numerical solution of systems of reaction-diffusion equations,

$$\partial_t u + a \cdot \nabla u = \Delta u + F(x,t,u), \quad x \in \Omega \subset \mathbf{R}^3, \, t > 0. \tag{1}$$

Here, u is a vector-valued function, $u \equiv u(x,t) \in \mathbf{R}^m$, m is large, and the corresponding system of ODEs, $\partial_t u = F(x,t,u)$, is stiff. Typical examples arise in air pollution studies, where a is the given wind field and the nonlinear function F models the atmospheric chemistry.

The time integration of Eq. (1) is best handled by the method of characteristics [1]. The problem is thus reduced to designing for the reaction-diffusion part a fast solver that has good stability properties for the given time step and does not require the computation of the full Jacobi matrix.

An operator-splitting technique, even a high-order one, combining a fast nonlinear ODE solver with an efficient solver for the diffusion operator is less effective when the reaction term is stiff. In fact, the classical Strang splitting method may underperform a first-order source splitting method [2]. The algorithm we propose in this paper uses an a posteriori filtering technique to stabilize the computation of the diffusion term. The algorithm parallelizes well, because the solution of the large system of ODEs is done pointwise; however, the integration of the chemistry may lead to load-balancing problems [3, 4]. The Tchebycheff acceleration technique proposed in [5] offers an alternative that complements the approach presented here.

To facilitate the presentation, we limit the discusssion to domains Ω that either admit a regular discretization grid or decompose into subdomains that admit regular discretization grids. We describe the algorithm for one-dimensional domains in Section 2 and for multidimensional domains in Section 3. Section 4 briefly outlines future work.

[1]CDCSP-ISTIL, Université Claude Bernard Lyon 1, cedex 69622 Villeurbanne, France

[2]Mathematics and Computer Science Division, Argonne National Laboratory, Argonne, IL 60439, USA. Work supported by the Mathematical, Information, and Computational Sciences Division subprogram of the Office of Advanced Scientific Computing Research, U.S. Department of Energy, under Contract W-31-109-Eng-38.

[3]Institute for Mathematical Modeling, Russian Academy of Science, 125047, Moscow, Russia. Supported by the Russian Foundation of Basic Research under Grant 00-01-0291.

2 One-Dimensional Domains

Consider the scalar equation

$$\partial_t u = \partial_x^2 u + f(u), \quad x \in (0,\pi),\, t > 0. \tag{2}$$

We combine a backward Euler approximation in time with an explicit finite-difference approximation of the diffusive term,

$$\frac{3u^{n+1} - 4u^n + u^{n-1}}{2\Delta t} = 2D_{xx}u^n - D_x u^{n-1} + f(u^{n+1}). \tag{3}$$

This scheme is second-order accurate in both space and time [6, 7, 8]. To analyze its stability, we take the Fourier transform of the linear equation,

$$\frac{3\hat{u}^{n+1} - 4\hat{u}^n + \hat{u}^{n-1}}{2\Delta t} = \Lambda_k(2\hat{u}^n - \hat{u}^{n-1}), \tag{4}$$

where $\Lambda_k = 2h^{-2}(\cos(hk) - 1)$, from which we obtain the stability condition

$$2\frac{\Delta t}{h^2}\left|\cos\left(\frac{k\pi}{N}\right) - 1\right| < \frac{4}{3}, \quad h = \frac{\pi}{N}. \tag{5}$$

Thus we conclude that the time step must satisfy the constraint

$$\Delta t < \tfrac{1}{3}h^2. \tag{6}$$

However, this constraint is imposed by the high frequencies, which are poorly handled by second-order finite differences anyway. For example, with central differences, the relative error for high-frequency waves $\cos(kx)$ with $k \approx N$ can grow at a rate of up to 9%. The idea is therefore to relax the constraint on the time step by applying a filter after each time step, which removes the high frequencies but maintains second-order accuracy in space.

2.1 Filters

By a *filter of order p* we mean an even function $\sigma : \mathbf{R} \to \mathbf{R}$ that satisfies the conditions (i) $\sigma(0) = 1$, (ii) $\sigma^{(l)}(0) = 0$ for $l = 1, \ldots, p-1$, (iii) $\sigma(\eta) = 0$ for $|\eta| \geq 1$, and (iv) $\sigma \in C^{p-1}(\mathbf{R})$.

Theorem [9]. Let f be a piecewise C^p function with one point of discontinuity, ξ, and let σ be a filter of order p. For any point $y \in [0, 2\pi]$, let $d(y) = \min\{|y - \xi + 2k\pi| : k = -1, 0, 1\}$. If $f_N^\sigma = \sum_{k=-\infty}^{\infty} \hat{f}_k \sigma(k/N) \mathrm{e}^{iky}$, then

$$|f(y) - k/N| \leq CN^{1-p}(d(y))^{1-p}K(f) + CN^{1/2-p}\|f^{(p)}\|_{L^2},$$

where

$$K(f) = \sum_{l=0}^{p-1}(d(y))^l |f^{(l)}(\xi^+) - f^{(l)}(\xi^-)| \int_{-\infty}^{\infty} |G_l^{(p-l)}(\eta)|\, \mathrm{d}\eta.$$

In other words, a discontinuity of f leads to a Fourier expansion with an error that is $O(1)$ near the discontinuity and $O(N^{-1})$ away from the discontinuity. We must therefore apply a shift and extend to $[0, 2\pi]$ before applying a filter.

2.2 The Algorithm

We now describe the postprocessing algorithm that is to be applied after each time step. (We do not explicitly indicate the dependence of u on the time step, and we use the abbreviations $u_0 = u(0)$ and $u_\pi = u(\pi)$.)

First, we apply a low-frequency shift,

$$v(x) = u(x) - (\alpha_1 + \alpha_2 \cos(x)), \quad \alpha_1 = \tfrac{1}{2}(u_0 - u_\pi),\ \alpha_2 = \tfrac{1}{2}(u_0 + u_\pi). \tag{7}$$

Then we extend v to $(0, 2\pi)$, using the definition

$$v(2\pi - x) = -v(x), \quad x \in (0, \pi). \tag{8}$$

Thus, v is a 2π-periodic function in $C^1(0, 2\pi)$. Let $\hat{v}_k$ be the kth coefficient of its Fourier expansion.

Next, we apply an eighth-order filter [9],

$$\sigma_N v(x) = \sum_k \sigma\left(\kappa \frac{k}{N}\right) \hat{v}_k \mathrm{e}^{ikx}, \tag{9}$$

where

$$\sigma(\xi) = (35 - 84y + 70y^2 - 20y^3)y^4, \quad y \equiv y(\xi) = \tfrac{1}{2}(1 + \cos(\pi\xi)). \tag{10}$$

Here, κ is a stretching factor, $\kappa > 1$. The correct choice of κ follows from a Fourier analysis of Eq. (5),

$$\kappa > \kappa_c = \frac{\pi}{\cos(1 - 2h^2/(3\Delta t))}. \tag{11}$$

The choice $\kappa = \frac{1}{2}\kappa_c$ gives satisfactory results, but in principle one can compute the optimum value of κ at each time step by monitoring the growth of the high-frequency waves that have not been completely filtered out.

Finally, we recover u from the inverse shift,

$$u(x) = \sigma_N v(x) + \alpha_1 + \alpha_2 \cos(x). \tag{12}$$

The theorem quoted in the preceding section shows that the filtering process may affect the spatial accuracy of the method. Since the filter is applied to a 2π-periodic function that is C^1 at the points $x_k = k\pi$, $k \in \mathbf{Z}$, and C^2 everywhere else, the error is of the order of N^{-2} in the neighborhood of x_k and N^{-3} away from x_k. In principle, we maintain therefore second-order accuracy in space as long as κ is of order one.

If the solution is in $C^3(0, \pi)$ at each time level, we can improve the algorithm by replacing the first-order shift (7) by a third-order shift,

$$v(x) = u(x) - \sum_{j=1}^{4} \alpha_j \cos((j-1)x), \tag{13}$$

such that the extension of v to a 2π-periodic function is in $C^3(0, 2\pi)$. The first- and third-order derivatives of v are zero at the points x_k, and the second-order derivative is approximately given by

$$u_{xx}(x_k) \approx \frac{3u^{n+1}(x_k) - 4u^n(x_k) + u^{n-1}(x_k)}{2\Delta t} - f(u^{n+1}(x_k)). \tag{14}$$

The coefficients α_j are found by solving a linear system of equations,

$$\begin{aligned} \alpha_1 + \alpha_2 + \alpha_3 + \alpha_4 &= u(0), \\ \alpha_1 - \alpha_2 + \alpha_3 - \alpha_4 &= u(\pi), \\ -\alpha_2 - 4\alpha_3 - 9\alpha_4 &= u_{xx}(0), \\ \alpha_2 - 4\alpha_3 + 9\alpha_4 &= u_{xx}(\pi). \end{aligned}$$

The third-order shift improves the performance of the filter for large κ and allows for a larger time step.

2.3 Numerical Results

Figure 1 shows some accuracy results for Eq. (2), where

$$u(x,t) = \cos(t)((x/\pi)^4 + \cos(3x)), \quad x \in (0,\pi),\, t > 0. \tag{15}$$

We observe a plateau for small time steps, when the second-order spatial er-

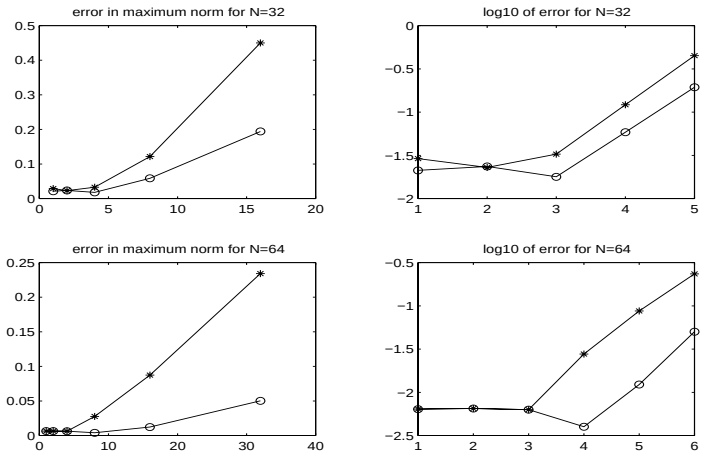

Figure 1: Accuracy of the stabilized explicit scheme for the heat equation. The horizontal coordinate is $3\Delta t/h^2$. * : first-order shift; o : third-order shift.

ror dominates. The second-order error in time becomes dominant as the time step increases. The figure confirms the superior performance of the third-order shift (13) over the first-order shift (7) at large time steps.

Although the algorithm is based only on linear stability considerations, it is still effective for systems of nonlinear reaction-diffusion equations. In Figure 2 we present some results for a predator-prey system,

$$\partial_t u = \partial_{xx} u + au - buv,\ \partial_t v = \partial_{xx} v - cu - duv, \quad x \in (0, \pi),\ t > 0, \tag{16}$$

with $a = 1.2$, $b = 1.0$, $c = 0.1$, and $d = 0.2$. At these parameter values, the

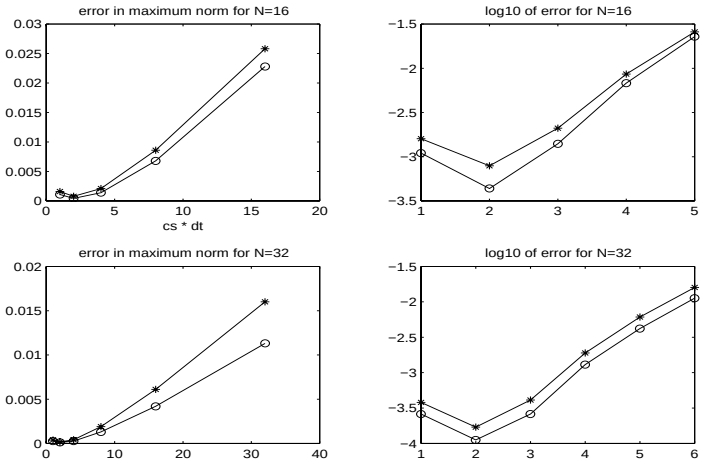

Figure 2: Accuracy of the stabilized explicit scheme for the predator-prey system. The horizontal coordinate is $3\Delta t/h^2$. * : first-order shift; o : third-order shift.

ODE system (reactions only) has a limit cycle. However, when the boundary conditions are constant in time, the solution of the reaction-diffusion system goes to steady state. To build a relevant test case for the algorithm, we impose periodic excitations at both boundaries,

$$u(0/\pi, t) = u_{0/\pi}(1 + \cos(t)),\ v(0/\pi, t) = v_{0/\pi}(1 + \cos(t)), \quad t > 0.$$

Although the time step can still be limited by nonlinear instabilities, we never observed negative values of the unknowns u and v, which are commonly associated with such instabilities.

The algorithm (7)–(12) extends in a straightforward way when one uses a domain-decomposition scheme with overlapping subdomains. One simply applies the same algorithm at each time step to each subdomain separately. However, the number of waves per subdomain is of the order of the total number of grid points, N, divided by the number of subdomains, N_d, so the balance between the order of accuracy of the filter— $(N/(\kappa N_d))^{-3}$ for a first-order filter or $(N/(\kappa N_d))^{-5}$ for a third-order filter—and the second-order accuracy N^{-2} of the spatial discretization of the underlying algorithm (3) deteriorates as N_d increases. The maximum time step for which the scheme remains stable may

become even less than when no domain decomposition is used. Furthermore, the Gibbs phenomenon tends to destabilize the algorithm. This phenomenon is a consequence of the jump in the derivatives at the endpoints of the subdomains (second-order derivatives in the case of the first-order shift (7), fourth-order derivatives in the case of the third-order shift (13)). Since the Gibbs phenomenon arises at the artificial interface and is damped away from it, an increase of the overlap generally produces a composite signal u that has fewer oscillations than each of the piecewise (overlapping) components. One can therefore obtain good results by adapting the size of the overlap. The larger the overlap, the larger the time step that can be taken; see Figure 3.

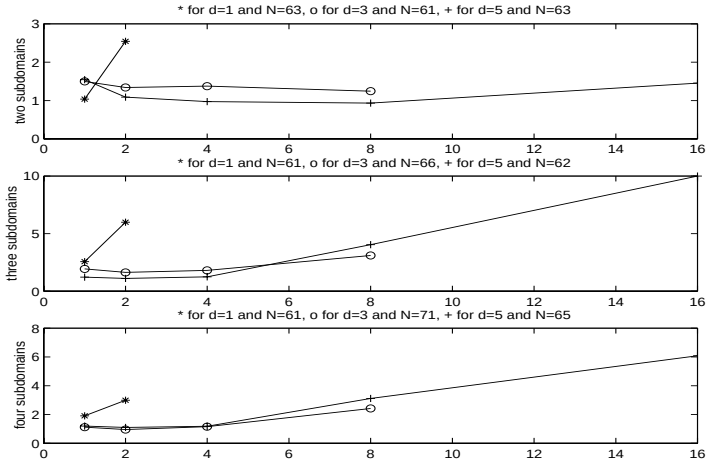

Figure 3: Accuracy of the stabilized explicit scheme applied to the heat equation with overlapping subdomains.

3 Two-Dimensional Domains

We now consider a Dirichlet problem in two dimensions,

$$\partial_t u = \Delta u + f(u), \quad (x,y) \in (0,\pi)^2,\, t > 0, \tag{17}$$

$$u(x,0/\pi) = g_{0/\pi}(x),\ u(0/\pi,y) = h_{0/\pi}(y), \quad x,y \in (0,\pi), \tag{18}$$

where the functions g satisfy the compatibility conditions $g_{0/\pi}(0) = h_0(0/\pi)$ and $g_{0/\pi}(\pi) = h_\pi(0/\pi)$. We consider a numerical scheme similar to Eq. (3), where the diffusive term is approximated, for example, by a five-point stencil. The postprocessing algorithm is essentially the same, except that we need an apropriate low-frequency shift so we can apply a filter to a smooth periodic function in two space dimensions. The shift is constructed in two steps. In the

first step, we render the boundary condition in the x direction homogeneous,

$$v(x,y) = u(x,y) - (\alpha_1(y) + \alpha_2(y)\cos(x)), \tag{19}$$

$$\alpha_1(y) = \tfrac{1}{2}(g_0(y) - g_\pi(y)),\ \alpha_2(y) = \tfrac{1}{2}(g_0(y) + g_\pi(y)). \tag{20}$$

In the second step, we shift in the y direction,

$$w(x,y) = v(x,y) - (\beta_1(x) + \beta_2(x)\cos(y)), \tag{21}$$

$$\beta_1(x) = \tfrac{1}{2}(v(x,0) - v(x,\pi)),\ \beta_2(x) = \tfrac{1}{2}(v(x,0) + v(x,\pi)). \tag{22}$$

The final step is the reconstruction step,

$$u(x,y) = \sigma_N w(x,y) + \alpha_1(y) + \alpha_2(y)\cos(x) + \beta_1(x) + \beta_2(x)\cos(y). \tag{23}$$

To make sure that no high-frequency waves remain, we filter the high-frequency components from the boundary conditions g with a procedure similar to (7)–(12).

It is much more difficult to construct a high-order filter similar to (13) in two dimensions, because the second-order derivatives cannot be obtained from the PDE, as in the one-dimensional case (14). So far, we have used only the first-order shifts (19) and (21) in our numerical experiments. Nevertheless, the algorithm allows for a significant increase of the time step. We have also tested the domain-decomposition version of the algorithm, using strip subdomains with an adaptive overlap, with good results.

We note that the computation in each block can be done in parallel and that the Jacobi matrix does not depend on the spatial variables. The arithmetic complexity of the algorithm is therefore relatively small. Also, the algorithm is suitable for multicluster architectures. Each block can be assigned to a cluster, and parallel fast sine transforms can be used for the filtering process inside each cluster. The cost of communication between blocks is minimal, since the scheme is similar to the communication scheme of the additive Schwarz algorithm.

4 Conclusion

In this paper we have presented a postprocessing algorithm that stabilizes the time integration of systems of reaction-diffusion equations when the diffusion term is treated explicitly. The algorithm is easy to code and can be combined with domain-decomposition methods that use regular grids in each subblock. In future work, we will consider the performance of its parallel implementation and its robustness for large systems of reaction-diffusion equations with stiff chemistry, which arise in some air pollution models.

References

[1] O. Pironneau. *Finite Element Methods for Fluids.* Wiley, 1989.

[2] J. G. Verwer and B. Sportisse. A note on operator splitting in a stiff linear case. Technical Report MAS-R9830, CWI, Amsterdam, Neth., 1998. http://www.cwi.nl.

[3] D. Dabdub and J. H. Seinfeld. Parallel computation in atmospheric chemical modeling. *Parallel Computing*, 22:111–130, 1996.

[4] H. Elbern. Parallelization and load balancing of a comprehensive atmospheric chemistry transport model. *Atmospheric Environment*, 31:3561–3574, 1997.

[5] J. J. Droux. *Simulation numérique bidimensionnelle et tridimensionnelle de processus de solidification*. PhD thesis, Lausanne EPFL, 1990. Thèse No. 901.

[6] L. Petzold. Automatic selection of methods for solving stiff and nonstiff systems of ordinary differential equations. *SIAM J. Sci. Stat. Comput.*, 4:136–148, 1983.

[7] A. Sandu, J. G. Verwer, M. van Loon, G. R. Carmichael, F. A. Potra, D. Dabdub, and J. H. Seinfeld. Benchmarking stiff ODE solvers for atmospheric chemistry problems–I: Implicit vs. explicit. *Atmospheric Environment*, 31:3151–3166, 1997.

[8] J. G. Verwer, W. H. Hundsdorfer, and J. G. Blom. Numerical time integration for air pollution models. Technical Report MAS-R9825, CWI, Amsterdam, Neth., 1998. http://www.cwi.nl.

[9] D. Gottlieb and Chi-Wang Shu. On the Gibbs phenomenon and its resolution. *SIAM Review*, 39:644–668, 1997.